Bayesian Wavelet and Fourier Transform Kernel Regression and Classification in RKHS

by

Xueying Zhang

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science

in

Statistical Machine Learning

Department of Mathematical and Statistical Sciences
University of Alberta

Abstract

Kernel methods are often used for nonlinear regression and classification in machine learning because they are computationally cheap and accurate. Fourier basis and wavelet basis are the bases that can efficiently approximate the kernel functions. In previous research, Bayesian approximate kernel regression with Fourier transform has been proposed [1]. With the proposed method, we use the analytic properties of the reproducing kernel Hilbert space (RKHS) to define a linear vector space that captures nonlinear structures. We map the data into a low-dimensional randomized feature space using Fourier transform and convert kernel function into operations of a linear machine. A Bayesian approximate kernel regression model is then formulated with the application of a generalized kernel model and the Bayesian method. We replace Fourier transform with wavelet transform in randomized feature space to approximate kernel functions. We formulate a new Bayesian approximate kernel model with wavelet transform and use the Gibbs sampler to compute the parameters of the model. We then make a comparison between the performance of Fourier-based and wavelet-based Bayesian approximate kernels solving both regression and classification problems.

Preface

This thesis is an original work by 'Xueying Zhang'. No part of this thesis has been previously published.

Acknowledgements

I would like to express my gratitude to everyone who helped me complete this thesis. I am most grateful to my supervisors, Dr.Yaozhong Hu and Dr.Linglong Kong. Their strong academic support, enlightening suggestions, patience, and encouragement helped me a lot during writing this thesis. Without their guidance and support, this work would be impossible. Also, I would like to thank Dr.Wenxing Guo, who helped me a lot during my research.

I must also express profound gratitude to my parents and friends for their continuous and unfailing support throughout my graduate study and this research. Thank you.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Machine learning problems with observations substantially smaller than the number of available variables, which are known as large p small n problem, are widespread and full of challenges. It is normal to use principal component analysis to lower the dimensions [2] or to use variable selection to reduce the number of variables [3]. Variable selection is well-developed for linear regression models [1], but it might not be practical or applicable in some situations. In this thesis, we focus on using nonlinear regression models to handle the large p small n problem in the reproducing kernel Hilbert spaces (RKHS) [4].

Since support vector machine (SVM) [5] was proposed, kernel supervised learning methods in RKHS have been widely used. Generalized kernel models [6] are extensions of the generalized linear models induced by a reproducing kernel in the feature space. A usual way to train a nonlinear support vector machine is to approximate the factorization of the kernel matrix and process the columns of the factor matrix as features in a linear machine [7]. In this thesis, we approximate the kernel function by factoring the kernel function itself [8] instead. This method maps high-dimensional data into low-dimensional randomized feature space [8].

Rahimi and Recht (2007) notice that the kernel in the models can be approximated by random Fourier features [8]. Inspired by their work, we propose their method to approximate the kernel functions using random wavelet features. Wavelet transform

is well localized in both time and frequency domains, while the Fourier transform is only localized in the frequency domain [9]. Our experiments indicate that random wavelet features yields higher accuracy in solving both classification and regression problems.

Bayesian approaches are applied to nonlinear classification and regression. Bayesian binary classification models in RKHS are proposed to analyze microarray data and produce smaller classification errors than some existing classification methods [10]. Bayesian approximate kernel regression model for nonlinear regression [1] performs well in genomic selection and association mapping. In this thesis, we mainly restrict ourselves to the Bayesian approximate kernel regression model's framework and apply it to classification problems.

In Chapter 2, we present the basic definitions of RKHS and define the penalized loss function in RKHS. In Chapter 3, we use random Fourier and wavelet features to approximate the kernel function. In Chapter 4, we formalize the Bayesian approximate kernel methods and apply it to classification and regression. In Chapter 5, we apply our methods to simulated data and real data, and compare the performance of Fourier-based and wavelet-based Bayesian approximate kernel models. Specifically, we focus on large p small n data. Finally, we make a conclusion in Chapter 6.

Chapter 2

Background

2.1 Basic Definition

In this section, we review the basic concepts related to reproducing kernel Hilbert space (RKHS). Then we model the high dimensional function $f(x)$ by adopting the RKHS approach.

Definition 2.1.1 (Norm). Let $\mathcal{H}$ be a vector space over $\mathbb{R}$. A function $\|\cdot\|_{\mathcal{H}} : \mathcal{H} \to [0, \infty)$ is called a *norm* on $\mathbb{R}$ if

1. $\|f\|_{\mathcal{H}} = 0$ if and only if $f = 0$
2. $\|\lambda f\|_{\mathcal{H}} = |\lambda| \|f\|_{\mathcal{H}}$, $\forall \lambda \in \mathbb{R}, \forall f \in \mathcal{H}$
3. $\|f + g\|_{\mathcal{H}} \leq \|f\|_{\mathcal{H}} + \|g\|_{\mathcal{H}}, \forall f, g \in \mathcal{H}$.

Definition 2.1.2 (Cauchy sequence). A sequence $\{x_n\}_{n=1}^{\infty}$ of elements of a normed vector space is called a *Cauchy sequence* if for every $\epsilon > 0$, there exists $N \in \mathbb{N}$ such that for all $m, n \geq N$, $\|x_m - x_n\|_{\mathcal{H}} < \epsilon$.

Definition 2.1.3 (Complete space). A space $\mathcal{M}$ is *complete* if every Cauchy sequence in $\mathcal{M}$ converges in $\mathcal{M}$.

Definition 2.1.4 (Banach space). A Banach space is a complete normed space.

Definition 2.1.5 (Inner product). Let $\mathcal{H}$ be a vector space over $\mathbb{R}$. A function $\langle \cdot, \cdot \rangle_{\mathcal{H}} : \mathcal{H} \times \mathcal{H} \to \mathbb{R}$ is defined as an *inner product* on $\mathcal{H}$ if

1. $\langle \lambda_1 f_1 + \lambda_2 f_2, g \rangle_{\mathcal{H}} = \lambda_1 \langle f_1, g \rangle_{\mathcal{H}} + \lambda_2 \langle f_2, g \rangle_{\mathcal{H}}$
2. $\langle f, g \rangle_{\mathcal{H}} = \langle g, f \rangle_{\mathcal{H}}$

3. $\langle f, f\rangle_{\mathcal{H}} \geq 0$ and $\langle f, f\rangle_{\mathcal{H}} = 0$ if and only if $f = 0$.

We can define a *norm* induced by the *inner product*:

$$\|f\|_{\mathcal{H}} = \langle f, f\rangle_{\mathcal{H}}^{1/2}$$

and they satisfy:

1. $|\langle f, g\rangle| \leq \|f\| \cdot \|g\|$
2. $\|f + g\|^2 + \|f - g\|^2 = 2\|f\|^2 + 2\|g\|^2$
3. $4\langle f, g\rangle = \|f + g\|^2 - \|f - g\|^2$.

Definition 2.1.6 (Hilbert space). A *Hilbert space* $\mathcal{H}$ is a complete inner product space where every Cauchy sequence converges to a limit.

Example 2.1.7. Let μ be a positive measure on $\mathcal{X} \subset \mathbb{R}^d$. The space $L_2(\mathcal{X}; \mu)$ is a Hilbert space with the inner product

$$\langle f, g\rangle = \int_X f(x)g(x)d\mu.$$

Definition 2.1.8 (Linear operator). Define a function $D : \mathcal{F} \to \mathcal{G}$ where $\mathcal{F}$ and $\mathcal{G}$ are the normed linear spaces in $\mathbb{R}$. D is defined as a *linear operator* if and only if it satisfies:

1. $D(\alpha f) = \alpha(Df) \quad \forall \alpha \in \mathbb{R}, f \in \mathcal{F}$
2. $D(f + g) = Df + Dg \quad \forall f \in \mathcal{F}, g \in \mathcal{G}$.

Definition 2.1.9 (Operator norm). The operator norm of a linear operator $D : \mathcal{F} \to \mathcal{G}$ is defined as

$$\|D\| = \sup_{f \in \mathcal{F}} \frac{\|Df\|_{\mathcal{G}}}{\|f\|_{\mathcal{F}}}.$$

Definition 2.1.10 (Continuous). A function $f : \mathcal{H} \to \mathcal{G}$ is called *continuous* at $x_0 \in \mathcal{H}$ if for every $\epsilon > 0$, there exists a $\delta > 0$ such that

$$\|x - x_0\|_{\mathcal{H}} < \delta \quad \text{implies} \quad \|f(x) - f(x_0)\|_{\mathcal{G}} < \epsilon.$$

Definition 2.1.11 (Kernel). Let $\mathcal{X}$ be a non-empty set. The function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is defined as a *Kernel* if there exists a real Hilbert space $\mathcal{H}$ and a map $\phi : \mathcal{X} \to \mathcal{H}$ such that $\forall x, y \in \mathcal{X}$,

$$k(x, y) = \langle \phi(x), \phi(y) \rangle_{\mathcal{H}}.$$

Such map $\phi : \mathcal{X} \to \mathcal{H}$ is defined as the feature map, and space $\mathcal{H}$ is defined as the feature space.

Example 2.1.12. Some well adopted nonlinear kernel functions include

1. Polynomial kernel: $k(x, y) = (x^T y + a)^b$, where $a \geq 0$ and $b \in \mathbb{N}$
2. Sigmoid kernel: $k(x, y) = \tanh(ax^T y + b)$
3. Gaussian kernel: $k(x, y) = \exp\left(-\frac{1}{2\sigma^2}\|x - y\|^2\right)$, where $\sigma > 0$.

The feature map ϕ can be seen from the example of polynomial kernel. Let $b = 2$, we have

$$\begin{aligned} k(x, y) &= \left(\sum_{i=1}^{n} x_i y_i + a \right)^2 \\ &= \sum_{i=1}^{n} \left(x_i^2\right)\left(y_i^2\right) + \sum_{i=2}^{n} \sum_{j=1}^{i-1} \left(\sqrt{2} x_i x_j\right)\left(\sqrt{2} y_i y_j\right) + \sum_{i=1}^{n} \left(\sqrt{2a} x_i\right)\left(\sqrt{2a} y_i\right) + a^2 \end{aligned}$$

Thus, $\phi(x)$ is given by

$$\phi(x) = \left\langle x_n^2, \cdots, x_1^2, \sqrt{2} x_n x_{n-1}, \cdots, \sqrt{2} x_2 x_1, \sqrt{2a} x_n, \ldots, \sqrt{2a} x_1, a \right\rangle$$

2.2 Reproducing Kernel Hilbert Space

There is an issue that for many well-adopted kernels, the dimension of the Hilbert space is infinite [11]. It is preferred to solve an optimization problem in a finite-dimensional space when training the dataset. We define a class of space called reproducing kernel Hilbert space (RKHS) that transfers the infinite-dimensional space to finite-dimensional space.

Definition 2.2.1. Let $\mathcal{X}$ be a set. A *reproducing kernel Hilbert space* over $\mathcal{X}$ is a Hilbert space $\mathcal{H}$ consisting of some functions on $\mathcal{X}$ such that for each $x \in \mathcal{X}$, there is a function $k_x \in \mathcal{H}$ with the property

$$\langle f, k_x \rangle_{\mathcal{H}} = f(x) \quad (\forall f \in \mathcal{H}),$$

where $k(\cdot, x) := k_x(\cdot)$ is called a *reproducing kernel* of $\mathcal{H}$.

Proposition 2.2.2. The reproducing kernel $k(x, y)$ is symmetric and positive definite: $k(x, y) = k(y, x)$ and for $x_1, \cdots, x_n \in \mathcal{X}$ and $a_1, \cdots a_n \in \mathcal{R}$

$$\sum_{i,j=1,\cdots,n} a_i a_j k\left(x_i, x_j\right) \geq 0.$$

Suppose we are given a set of training data $\{(x_i, y_i)\}_{i=1}^n$ where $x_i \in \mathcal{X} \subseteq \mathbb{R}^p$ is an input vector and $y_i \in \mathcal{Y} \subseteq \mathbb{R}$ is the continuous output for a regression problem or $y_i = \pm 1$ is the binary output for a classification problem. Consider the standard non-parametric problem and estimate $f(x)$ by the following penalized loss function [12]

$$\widehat{f} = \arg\min_{f \in \mathcal{H}} \left[\frac{1}{n} \sum_{i=1}^{n} L\left(f\left(\mathbf{x}_i\right), y_i\right) + \lambda \|f\|_{\mathrm{K}}^2 \right], \tag{2.1}$$

where $L(f(\mathbf{x}), y)$ is a loss function. $\|f\|_{\mathrm{K}}^2$ is the RKHS norm.

Theorem 2.2.3 (Nonparametric Representer theorem). Let $\mathcal{X}$ be a non-empty set, k is a positive definite real-valued kernel on $\mathcal{X} \times \mathcal{X}$, g is a strictly monotonically increasing real-valued function on $[0, \infty]$, L is an arbitrary cost function and $\mathcal{F}$ is a class of functions that is given by [13]

$$\mathcal{F} = \left\{ f \in \mathbf{R}^{\mathcal{X}} \mid f(\cdot) = \sum_{i=1}^{\infty} \beta_i k\left(\cdot, z_i\right), \beta_i \in \mathbf{R}, z_i \in \mathcal{X}, \|f\|_K < \infty \right\},$$

where $\|\cdot\|_K$ is the norm in the RKHS. Then for any $f \in \mathcal{F}$ minimizing the penalized loss function

$$L\left(\left(x_1, y_1, f\left(x_1\right)\right), \ldots, \left(x_n, y_n, f\left(x_m\right)\right)\right) + g(\|f\|_K)$$

admits a representation of the form

$$f(\cdot) = \sum_{i=1}^{n} \alpha_i k\left(\cdot, x_i\right).$$

By the representer theorem, the solution for (2.1) can be written as

$$\widehat{f}(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i k\left(\mathbf{x}, \mathbf{x}_i\right),$$

where $\alpha = \{\alpha_i\}_{i=1}^{n}$ are the corresponding kernel coefficients.

Notice that $\|f\|_{\mathrm{K}}^2 = \sum_{i,j=1}^{n} k\left(\mathbf{x}_i, \mathbf{x}_j\right) \alpha_i \alpha_j$, substituting it into (2.1) we obtain [6]

$$\widehat{f} = \arg\min_{f \in \mathcal{H}} \left[\frac{1}{n} \sum_{i=1}^{n} L\left(f\left(\mathbf{x}_i\right), y_i\right) + \lambda \alpha' k \alpha \right],$$

where $\alpha = (\alpha_1, \cdots, \alpha_n)'$ is an $n \times 1$ regression vector and $k = (k_1, \cdots k_n)$ is the $n \times n$ kernel matrix with $k_i = (k(x_i, x_1), \cdots, k(x_i, x_n))'$

Chapter 3

Random features

3.1 Random features

The kernel methods can be used to generate features for algorithms easily. It is based on the inner product between pairs of input points [8]. However, when dealing with large training sets, the kernel methods consume a substantial amount of computational and storage resources. Instead of using the normal kernel function, we introduce a randomized feature map $z : \mathbb{R}^p \to \mathbb{R}^d$ that maps the data into a low-dimensional inner product space. It utilizes the inner product between a pair of transformed points to approximate the kernel function:

$$k(\mathbf{x}, \mathbf{y}) = \langle \phi(\mathbf{x}), \phi(\mathbf{y}) \rangle \approx \mathbf{z}(\mathbf{x})'\mathbf{z}(\mathbf{y}),$$

where $x \in \mathbb{R}^p$ and $z(x) \in \mathbb{R}^d$.

With the kernel methods, evaluating the machine at a test point x requires computing $f(x) = \sum_{i=1}^{n} \alpha_i k(x, x_i)$, which has a time complexity of $O(np)$. For large datasets, the scaling of this kernel method is at least quadratic in the number of examples [14]. Therefore, this method is impractical if the dataset is beyond 10^4 elements.

After introducing the randomized feature map and learning a hyperplane w, a linear machine can be evaluated by simply computing $f(x) = w'z(x)$. With the randomized feature maps presented, the computation requires only $O(p+d)$ operations and storage. [8]. We can transform the input x with the low-dimensional z and apply

linear methods to approximate the nonlinear kernel machine at high speed.

Theorem 3.1.1 (Mercer-Hilbert-Schmidt Theorem) [11]. Let $\{\phi_j\}$ be an orthogonal sequence of continuous eigenfunctions on $L_2(\mathcal{X})$ and eigenvalues $l_1 \geq l_2 \geq \cdots \geq 0$. Let k be a continuous kernel on compact metric space $\mathcal{X}$, then $\forall x, y \in \mathcal{X}$

$$k(x, y) = \sum_{j=1}^{r} l_j \phi_j(x) \phi_j(y).$$

We define the feature functions $\psi(x) = \{\sqrt{l_j}\phi_j(x)\}_{j=1}^{r}$, i.e. $\psi_j(x) = \sqrt{l_j}\phi_j(x)$. Consequently, the estimated function f can be expressed as follows [6],

$$f(\mathbf{x}) = \sum_{j=1}^{r} b_j \psi_j(\mathbf{x}) = \psi(\mathbf{x})'\mathbf{b},$$

where $\mathbf{b} = (b_1, \ldots, b_r)'$.

Since there is the possibility that the r is infinite, we need to keep the first n $\psi_j(x)$ and set the remaining b_j,$j > n$ equal to zero [15]. Now we can use the finite-dimensional approximation.

Let $b = \psi\alpha$. From $f(x) = \sum_{i=1}^{n} \alpha_i k(x, x_i)$, we get $k = \psi'\psi$. For the shift-invariant kernel function: $k(x_i, y_j) = k(x_i - y_j)$, we have

$$k(x_i - x_j) = \psi'\psi \approx \mathbf{z}(\mathbf{x_i})'\mathbf{z}(\mathbf{x_j}) = \tilde{k}(x_i - x_j),$$

where $\tilde{k}$ is the approximate kernel.

To be more explicit, we represent $\mathbf{z}$ as $\tilde{\psi}$ and specify a matrix $\widetilde{\Psi} = \left[\tilde{\psi}(\mathbf{x}_1), \ldots, \tilde{\psi}(\mathbf{x}_n)\right]$ with a corresponding approximate kernel matrix [1]

$$\widetilde{K} = \widetilde{\Psi}^{\top}\tilde{\Psi}.$$

3.2 Random Fourier Features

Bochner's theorem plays an important role in the random Fourier features method. The theorem connects the positive definite kernel and Fourier transform. [8, 14, 16]

Theorem 3.2.1 (Bochner's theorem) [17]. Let f be a positive definite function on Y such that f is continuous at e and $f(e) = 1$, e is the identity element of Y. There exists a unique probability measure μ on Borel σ-field of X such that

$$f(y) = \int_X \langle x, y \rangle d\mu(x), \quad y \in Y$$

i.e. f is the Fourier transform of a unique probability measure μ on Y, where X is a locally compact abelian group and Y is a dual group.

Applying Bochner's theorem to the shift-invariant kernel, we have:

Theorem 3.2.2 [8]. A continuous kernel $k(x_i, x_j) = k(x_i - x_j)$ on $\mathbb{R}^p$ is positive definite if and only if $k(x_i - x_j)$ is the Fourier transform of a non-negative measure.

If $k(x_i - x_j)$ is in proper scale, then its Fourier transform $p(\boldsymbol{\omega})$ is a probability density function. Defining the shift-invariant kernel functions $k(x_i - x_j)$, the probability density $p(\boldsymbol{\omega})$ and $\eta_\omega(\mathbf{x}_i) = exp(\iota \boldsymbol{\omega}^T x_i)$ where $\iota^2 = -1$, we have the following Fourier expansion

$$\begin{aligned} k(x_i - x_j) &= \int_{\mathbb{R}^p} p(\boldsymbol{\omega}) \exp\left\{\iota \boldsymbol{\omega}^\top (x_i - x_j)\right\} d\boldsymbol{\omega} \\ &= \mathbb{E}_\omega \left[\eta_\omega(x_i) \eta_\omega(x_j)^*\right]. \end{aligned}$$

Therefore, $\eta_\omega(\mathbf{x}_i) \eta_\omega(\mathbf{x}_j)^*$ is an unbiased estimate of $k(x_i, x_j)$. Since probability density $p(\boldsymbol{\omega})$ and $k(x_i, x_j)$ are real values, we replace $\exp(\iota \boldsymbol{\omega}^T (x_i - x_j))$ with $\cos \boldsymbol{\omega}^T (x_i - x_j)$. Then we have the real-valued mapping

$$\tilde{\psi}_\omega(x_i) = \sqrt{2} \cos(\boldsymbol{\omega}^T x_i + b),$$

where $\boldsymbol{\omega}$ is drawn from $p(\boldsymbol{\omega})$ and b is drawn from $U[0, 2\pi]$, satisfying the condition

$$E\left[\tilde{\psi}_\omega(x_i)' \tilde{\psi}_\omega(x_j)\right] = k(x_i, x_j).$$

We can reduce the variance of $\tilde{\psi}_\omega(x_i)' \tilde{\psi}_\omega(x_j)$ by concatenating d randomly chosen $\tilde{\psi}_\omega$ into a column vector $\tilde{\psi}$ and normalizing each component by dividing them by $\sqrt{d}$ [8]. We apply the Monte Carlo method to random Fourier features and the approximation is as follows:

$$\omega_\ell \overset{\text{iid}}{\sim} p(\boldsymbol{\omega}), \quad \mathrm{b}_\ell \overset{\text{iid}}{\sim} U[0, 2\pi], \quad \ell = 1, \ldots, d$$

$$\boldsymbol{\Omega} = [\boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_d] \in \mathbb{R}^{p \times d}, \quad \mathbf{b} = [\mathrm{b}_1, \ldots, \mathrm{b}_d] \in \mathbb{R}^d$$

$$\tilde{\boldsymbol{\psi}}(\mathbf{x}_i)^\top = \sqrt{\frac{2}{d}} \cos(\mathbf{x}_i \boldsymbol{\Omega} + \mathbf{b}).$$

Let $\widetilde{\Psi} = \left[\tilde{\psi}(\mathbf{x}_1), \ldots, \tilde{\psi}(\mathbf{x}_n)\right]$, the approximate kernel martrix $\tilde{K} = \tilde{\Psi}^T \tilde{\Psi}$ converges to the exact kernel as the random sample size d goes to infinity [1].

3.3 Random Wavelet Features

The motivation of wavelet analysis is to approximate a signal or a function by using a mother wavelet function ψ

$$\psi_{a,b}(x) = |a|^{-1/2} \psi\left(\frac{x-b}{a}\right), \tag{3.1}$$

where $a, b \in \mathbb{R}$,$a \neq 0$, a is a dilation factor and b is a translation factor.

If $|a| < 1$, $\psi_{a,b}(x)$ has smaller time-width than $\psi(x)$ and is in a higher frequency; if $|a| > 1$, $\psi_{a,b}(x)$ has larger time-width than $\psi(x)$ and is in a lower frequency. Thus wavelet has time-widths adapted to their frequencies [9].

If a function $f(x) \in L_2(\mathbb{R})$, the wavelet transform of f is written as

$$\sum_{j \in Z} \sum_{k \in Z} \langle f, \psi_{j,k} \rangle \psi_{j,k}(t).$$

The wavelet coefficients of f is

$$\langle f, \psi_{j,k} \rangle = d_{j,k} = \int_{-\infty}^{\infty} f(t) \psi_{j,k}(t) dt.$$

Advantage of Wavelet transform [9]

- Wavelet transform is well localized in both the time and frequency domain.
- Wavelet transform is fast to compute.

- Given a function f, we can use a few coefficients to approximate it and achieve good results with wavelet transform.

- We can compress or denoise a signal without appreciable degradation.

If $\psi(x)$ is a mother wavelet and $x, x' \in \mathbb{R}^d$ then the dot-product wavelet kernel is

$$k(x, x') = \prod_{i=1}^{d} \left(\psi\left(\frac{x_i - b}{a}\right) \cdot \psi\left(\frac{x_i' - b}{a}\right) \right).$$

The translation-invariant wavelet kernel is [18]

$$k(x, x') = \prod_{i=1}^{d} \left(\psi\left(\frac{x_i - x_i'}{a}\right) \right).$$

Definition 3.3.1(Mercer's condition) A real-valued function $k(x, y)$ is said to fulfill Mercer's condition if for all square-integrable functions $g(x)$, we have [19]

$$\iint g(x)k(x, y)g(y)dxdy \geq 0.$$

If the Mercer's condition holds, we can write $k(x, y)$ as a dot product $k(x, y) = \langle \phi(\mathbf{x}) \cdot \phi(\mathbf{y}) \rangle$ [20].

Lemma 3.3.2 The dot-product wavelet kernel satisfies Mercer's condition, i.e. it can be written as a dot product.

Proof. For $\forall g(x) \in L_2(\mathbb{R}^d)$, we have

$$\begin{aligned}
&\iint_{L_2 \otimes L_2} k(\mathbf{x}, \mathbf{x}') g(\mathbf{x}) g(\mathbf{x}') d\mathbf{x} d\mathbf{x}' \\
&= \int_{L_2} \prod_{i=1}^{d} \psi\left(\frac{x_i - b}{a}\right) g(\mathbf{x}) d\mathbf{x} \int_{L_2} \prod_{i=1}^{d} \psi\left(\frac{x_i' - b}{a}\right) g(\mathbf{x}') d\mathbf{x}' \\
&= \left(\int_{L_2} \prod_{i=1}^{d} \psi\left(\frac{x_i - b}{a}\right) g(\mathbf{x}) d\mathbf{x} \right)^2 \geq 0
\end{aligned}$$

Thus the dot-product wavelet kernel can be represented as

$$\begin{aligned} k\left(x, x'\right) &= \prod_{i=1}^{d}\left(\psi\left(\frac{x_i - b}{a}\right) \cdot \psi\left(\frac{x_i' - b}{a}\right)\right) \\ &= \prod_{i=1}^{d}\psi\left(\frac{x_i - b}{a}\right) \cdot \prod_{i=1}^{d}\psi\left(\frac{x_i' - b}{a}\right) \\ &= \left\langle \Psi(x) \cdot \Psi\left(x'\right)\right\rangle, \end{aligned}$$

where $\Psi(x) = \prod_{i=1}^{d}\psi\left(\frac{x_i - b}{a}\right)$, $\Psi(x') = \prod_{i=1}^{d}\psi\left(\frac{x'_i - b}{a}\right)$.

We then can use the dot product of $\Psi(x)$ and $\Psi(x')$ to approximate the kernel function $k(x, x')$. Selecting a proper mother wavelet ψ is important for the random wavelet features method. Since we mainly focus on using random wavelet features to approximate the Gaussian kernel in this article, we choose the Morlet wavelet kernel function which decays as a Gaussian [21]. The mother wavelet of Morlet wavelet function is defined as

$$\psi(x) = \cos(1.75 * x)\exp\left(-x^2/2\right).$$

Therefore, we have

$$\begin{aligned} \psi(\frac{x-b}{a}) &= \cos(1.75 * (\frac{x-b}{a}))\exp\left(-(\frac{x-b}{a})^2/2\right) \\ &= \cos(1.75 * (mx - n))\exp\left(-(mx-n)^2/2\right), \end{aligned}$$

where $m = \frac{1}{a}$, $n = \frac{b}{a}$.

The Morlet wavelet kernel function can be written as

$$\begin{aligned} k(x, x') = &\prod_{i=1}^{d}\cos(1.75 * (mx_i - n))\exp\left(-(mx_i - n)^2/2\right) \cdot \\ &\prod_{i=1}^{d}\cos(1.75 * (mx_i' - n))\exp\left(-(mx_i' - n)^2/2\right). \end{aligned}$$

The approximation of Gaussian kernel using random wavelet features is formulated as follows:

$$m_\ell \overset{\text{iid}}{\sim} N(0,1), \quad n_\ell \overset{\text{iid}}{\sim} N(0,1), m_\ell, n_\ell \quad \text{are independent}, \quad \ell = 1, \ldots, d$$

$$\boldsymbol{M} = [m_1, \ldots, m_d] \in \mathbb{R}^{p \times d}, \quad \mathbf{n} = [\mathrm{n}_1, \ldots, \mathrm{n}_d] \in \mathbb{R}^d$$

$$\tilde{\psi}(x_j)^\top = (\prod_{i=1}^{d} \psi\left(m_1 x_{1i} - n_1\right), \cdots, \prod_{i=1}^{d} \psi\left(m_d x_{di} - n_d\right)), \quad j = 1, \cdots, n$$

Let $\widetilde{\Psi} = \left[\tilde{\psi}\left(\mathbf{x}_1\right), \ldots, \tilde{\psi}\left(\mathbf{x}_n\right)\right]$, we then can use $\tilde{K} = \tilde{\Psi}^T \tilde{\Psi}$ to approximate the kernel function, where $\psi(x)$ is the mother wavelet of Morlet wavelet function.

Chapter 4

Bayesian approximate kernel methods

4.1 Generalized Kernel Models

We can treat the loss function $L\left(f\left(\mathbf{x}_i\right), y_i\right)$ in (2.1) as a negative conditional log-likelihood using the logarithmic scoring rule [22]. The generalized linear model (GLM) [23] is defined as

$$y \sim p(y \mid \mu) \quad \text{with} \quad \mu = g^{-1}(X\beta),$$

where μ is the expected value of response y conditional on the input X, $X\beta$ is the linear operator and g is the link function.

The generalized kernel model (GKM) [6] is derived from the GLM and can be written as

$$y \sim p(y \mid \mu) \quad \text{with} \quad \mu = g^{-1}\left(\tilde{K}'\alpha\right). \tag{4.1}$$

This model can be obtained from the model

$$y \sim p(y \mid \mu) \quad \text{with} \quad \mu = \tau\left(\tilde{\psi}'\mathbf{b}\right),$$

where $\alpha = \tilde{K}^{-1}\tilde{\psi}'\mathbf{b}$.

The generalized models have been widely used in classification and regression problems which are based on kernel methods [24]. We can specify a proper likelihood and

link function depending on different applications. To be specific, the likelihood is set to be normal distribution and the link function is set to be the uniform distribution in the regression problems [1]. In this thesis, we apply these generalized kernel models to regression and classification problems.

Since the approximate kernel matrix $\tilde{K}$ is symmetric and positive definite, the spectral decomposition of $\tilde{K}$ is as follows,

$$\tilde{K} = \tilde{Q}\tilde{\Lambda}\tilde{Q}^T,$$

where $\tilde{Q}$ is an $n \times n$ orthogonal matrix whose ith column is the eigenvector q_i of $\tilde{K}$ and $\tilde{\Lambda} = Diag(\lambda_1, \cdots, \lambda_n)$ is a diagonal matrix, where eigenvalues $\lambda_1 \geq \lambda_2, \cdots, \geq \lambda_n$.

We rewrite the equation (4.1) as

$$y \sim p(y \mid \mu) \quad \text{with} \quad g^{-1}(\mu) = \widetilde{Q}\theta, \tag{4.2}$$

where $\theta = \widetilde{\Lambda}\tilde{Q}^\top\alpha$.

Eigenvectors corresponding to small eigenvalues can be truncated to reduce the computational complexity. Thus, we can keep the top s eigenvalues and consider $\tilde{Q}$ as an $n \times s$ matrix and $\tilde{\lambda}$ as an $s \times s$ diagonal matrix. We can further reduce the dimension from n to s parameters. This new representation can substantially speed up the process of estimating the model parameters, especially when n is large.

4.2 Projection onto Explanatory Variables

The standard projection operation is

$$\text{Proj}(X, y) = X^\dagger y,$$

where $X^\dagger = (X^T X)^{-1} X^T$ is the Moore–Penrose generalized inverse. For the Bayesian approach, priors over the parameters β induce the distribution on the projection Proj(X, y).

Consider a nonlinear function $E[y] = f = [f(x_1), \cdots, f(x_n)]$, we define the projection $\tilde{\beta}$ as

$$\tilde{\beta} = Proj(X, f).$$

Recall that $\alpha = \tilde{K}^{-1}\tilde{\psi}^T b$ and $\theta = \tilde{\Lambda}\tilde{Q}^T\alpha$, we have the following representation

$$b = (\tilde{\Lambda}\tilde{Q}^T\tilde{K}^{-1}\tilde{\psi^T})^{-1}\theta.$$

The projection of $f = \tilde{\psi}^T b$ can be written as

$$\tilde{\beta} = \mathbf{X}^{\dagger}\tilde{\psi}^T b.$$

4.3 Bayesian Hierarchical Model and Gibbs Sampler

Before specifying the Bayesian approximate kernel models, we first review some basic concepts in the applied Bayesian statistics. Bayes' original theorem is applied to probability mass functions, which is stated as

$$p(B \mid A) = \frac{p(A \mid B)p(B)}{p(A)},$$

where A and B are events and $P(B) \neq 0$.

However, we use probability distributions more frequently than probability mass functions in the application of the Bayesian theorem [25]. We then introduce Bayes' theorem for probability distributions,

$$f(\vartheta \mid X) = \frac{f(X \mid \vartheta)f(\vartheta)}{f(X)},$$

where $f(\vartheta \mid X)$ is the posterior distribution, $f(X \mid \vartheta)$ is the sample density, $f(\vartheta)$ is the prior distribution, ϑ is the parameter and X is the data.

The marginal probability of the data X is

$$f(X) = \int f(X \mid \vartheta)f(\vartheta)d\vartheta.$$

Since the sample density is proportional to the likelihood function, the posterior distribution is proportional to

$$p\left(\vartheta \mid \{\mathrm{y}_i, \mathbf{x}_i\}_{i=1}^n\right) \propto \exp\left\{-\sum_{i=1}^n L\left(f\left(\mathbf{x}_i\right), \mathrm{y}_i\right)\right\} \pi(\vartheta),$$

where $\pi(\vartheta)$ is the prior distribution and $\exp\left\{-\sum_{i=1}^n L\left(f\left(\mathbf{x}_i\right), \mathrm{y}_i\right)\right\}$ is the likelihood function.

Let x_j be an observation and ϑ_j is a parameter governing the data generating process for x_j. Assume that the parameters $\vartheta_1, \vartheta_2, \cdots, \vartheta_j$ are generated from the distribution governed by a hyperparameter φ. The Bayesian hierarchical model includes the following stages [26]

Stage 1. $x_j \mid \vartheta_j, \varphi \sim p(x_j \mid \vartheta_j, \varphi)$

Stage 2. $\vartheta \mid \varphi \sim p(\vartheta \mid \varphi)$

Stage 3. $\varphi \sim p(\varphi)$.

Thus, the posterior distribution is proportional to

$$\begin{aligned} p(\varphi, \vartheta_j \mid x_j) &\propto p(x_j \mid \vartheta_j, \varphi) p(\vartheta_j, \varphi) \\ &\propto p(x_j \mid \vartheta_j) p(\vartheta_j \mid \varphi) p(\varphi). \end{aligned}$$

Markov chain Monte Carlo (MCMC) methods include a class of algorithms for sampling from probability distributions. The development of MCMC methods allows us to compute the large Bayesian hierarchical model with thousands of unknown parameters [27]. In the application of the Bayesian hierarchical model, the Gibbs sampler is the most basic MCMC method. [25]. A general Gibbs sampler follows the following iterative process,

0. Assign a vector of starting values S, $\theta^{j=0} = S$, where j is the iteration count.

1. Let j= j+1.

2. Sample $\left(\theta_1^j \mid \theta_2^{j-1}, \theta_3^{j-1} \cdots, \theta_k^{j-1}\right)$.

3. Sample $\left(\theta_2^j \mid \theta_1^j, \theta_3^{j-1} \cdots, \theta_k^{j-1}\right)$.

k. Sample $\left(\theta_k^j \mid \theta_1^j, \theta_2^j, \cdots, \theta_{k-1}^j\right)$.

k+1. Return to step 1.

4.4 Bayesian approximate kernel method for regression

We restate equation (4.1) for the regression problem as,

$$y = \widetilde{K}\alpha + \varepsilon, \quad \varepsilon \sim \mathcal{N}\left(0, \tau^2 I\right), \tag{4.3}$$

where ε is the random error, $\mathcal{N}(\cdot, \cdot)$ is the multivariate normal distribution with the mean zero and the variance τ^2, and I is the identity matrix.

Combining the hierarchical model with the factor representation in equation (4.2), we can formulate the specific hierarchical model for the nonlinear regression model as follows [1],

$$\begin{aligned} y &= \widetilde{Q}\theta + \varepsilon, \quad \varepsilon \sim \mathcal{N}\left(0, \tau^2 I\right) \\ \theta &\sim \mathcal{N}\left(0, \sigma^2 \widetilde{\Lambda}\right) \\ \sigma^2, \tau^2 &\sim \text{ Scale-inv- } \chi^2(\nu, \phi). \end{aligned} \tag{4.4}$$

The idea of using θ in (4.4) instead of using α in (4.3) is from the Silverman g-prior [6]. The variance of random error τ^2 and the shrinkage parameter σ^2 both come from the scaled inverse chi-squared distribution with the degrees of freedom ν and the scale parameter ϕ. The probability density function of the scaled inverse chi-squared distribution over the domain $x > 0$ is

$$f\left(x; \nu, \phi\right) = \frac{\left(\phi\nu/2\right)^{\nu/2}}{\Gamma(\nu/2)} \frac{\exp\left[\frac{-\nu\phi}{2x}\right]}{x^{1+\nu/2}}.$$

Given the Bayesian hierarchical model in (4.4), we can propose the conditional densities $p(\theta \mid \sigma^2, \tau^2, y)$ using the Bayes' Theorem. To be specific,

$$\begin{aligned} p(\theta \mid \sigma^2, \tau^2, y) &\propto p(y \mid \theta, \tau^2) p(\theta \mid \sigma^2) \\ &\propto \mathcal{N}(m^*, n^*), \end{aligned}$$

where $n^* = \tau^2\sigma^2(\tau^2\widetilde{\Lambda}^{-1} + \sigma^2 I_q)^{-1}$ and $m^* = \tau^2 n^* \widetilde{Q}^T y$.

Similarly, we can propose the conditional densities for σ^2 and τ^2. We then use a Gibbs sampler to generate the joint posterior $p(\theta, \sigma^2, \tau^2 \mid y)$ and the procedures are as follows [1],

1. $\theta \mid \sigma^2, \tau^2, y \sim \mathcal{N}(m^*, n^*)$, with $n^* = \tau^2\sigma^2(\tau^2\widetilde{\Lambda}^{-1} + \sigma^2 I_q)^{-1}$ and $m^* = \tau^2 n^* \widetilde{Q}^T y$;
2. $\widetilde{\beta} = X^{\dagger}\tilde{\Psi}^{\top}\left(\widetilde{\Lambda}\tilde{Q}^{\top}\widetilde{K}^{-1}\widetilde{\Psi}^{\top}\right)^{-1}\theta$;
3. $\sigma^2 \mid \theta, \tau^2, y \sim$ Scale-inv $-\chi^2\left(v_\sigma^*, \phi_\sigma^*\right)$, where $v_\sigma^* = v+q$ and $\phi_\sigma^* = v_\sigma^{*-1}\left(v\phi + \theta^{\top}\tilde{\Lambda}^{-1}\theta\right)$;
4. $\tau^2 \mid \theta, \sigma^2, y \sim$ Scale-inv $-\chi^2\left(v_\tau^*, \phi_\tau^*\right)$, where $v_\tau^* = v+n$ and $\phi_\tau^* = v_\tau^{*-1}\left(v\phi + e^{\top}e\right)$,

with $e = y - \widetilde{Q}\theta$.

We can achieve the following set of posterior samples by repeating the above procedure for T times

$$\left\{\theta^{(t)}, \sigma^{2(t)}, \tau^{2(t)}, \widetilde{\beta}^{(t)}\right\}_{t=1}^{T}.$$

For the sample test X observed, the prediction is stated as

$$y = X\tilde{\beta}.$$

4.5 Bayesian approximate kernel method for classification

We extend the Bayesian approximate kernel method to binary classification. We also use the generalized kernel model for the classification problem,

$$y \sim p(y \mid \mu) \quad \text{with} \quad \mu = g^{-1}\left(\tilde{K}'\alpha\right).$$

We can specify the hierarchical model for classification using the factor representation where $\tilde{K} = \tilde{Q}\tilde{\Lambda}\tilde{Q}^T$

$$
\begin{aligned}
\mathrm{y}_i &= \begin{cases} 1 & \text{if} \quad \mathrm{s}_i > 0 \\ 0 & \text{if} \quad \mathrm{s}_i \leq 0 \end{cases} \\
s &= \widetilde{Q}\theta + \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, I), \\
\theta &\sim \mathcal{N}\left(0, \sigma^2 \widetilde{\Lambda}\right) \\
\sigma^2 &\sim \text{ Scale-inv- } \chi^2(\nu, \phi).
\end{aligned}
$$

The vector of latent responses is defined as $s = [\mathrm{s}_1, \cdots, \mathrm{s}_n]^\top$. The MCMC procedure here is similar to the posterior sampling of probit regression [28]. Posterior samples are generated by iterating the following procedures:

(1) For $i = 1, \ldots, n$

$$
\mathrm{s}_i^{(t+1)} \mid \theta, \sigma^2, s^{(t)}, y \sim \begin{cases} \mathrm{N}\left(\tilde{q}_i^\top \theta, 1\right) \mathbb{I}\left(\mathrm{s}_i^{(t)} \leq 0\right) & \text{if } \mathrm{s}_i^{(t)} \leq 0 \\ \mathrm{N}\left(\tilde{q}_i^\top \theta, 1\right) \mathbb{I}\left(\mathrm{s}_i^{(t)} > 0\right) & \text{if } \mathrm{s}_i^{(t)} > 0 \end{cases};
$$

(2) $\theta \mid s, \sigma^2, y \sim \mathcal{N}\left(m^*, n^*\right)$ where $n^* = \sigma^2 \left(\widetilde{\Lambda}^{-1} + \sigma^2 I\right)^{-1}$ and $m^* = n^* \widetilde{Q^\top} s$;

(3) $\widetilde{\beta} = X^\dagger \widetilde{\Psi}^\top \left(\widetilde{\Lambda} \tilde{Q}^\top \widetilde{K}^{-1} \widetilde{\Psi}^\top\right)^{-1} \theta$;

(4) $\sigma^2 \mid s, \theta, y \sim$ Scale-inv- $\chi^2\left(\nu^*, \phi^*\right)$, where $\nu^* = \nu + q$ and $\phi^* = \nu^{*-1}\left(\nu\phi + \theta^\top \widetilde{\Lambda}^{-1} \theta\right)$.

We obtain the following set of posterior samples by repeating the above procedure T times

$$
\left\{\theta^{(t)}, \sigma^{2(t)}, \widetilde{\beta}^{(t)}\right\}_{t=1}^{T}.
$$

For the observations X, the prediction holds $y = X\tilde{\beta}$. For each response y_i

$$
\mathrm{y}_i = \begin{cases} 1 & \text{if} \quad \mathrm{y}_i > 0 \\ 0 & \text{if} \quad \mathrm{y}_i \leq 0 \end{cases}
$$

Chapter 5

Numerical studies

In this chapter, we conduct simulations to evaluate the performances of random Fourier features and random wavelet features approximating the kernel function in both regression and classification problems. We also compare the Bayesian approximate kernel methods with other classical methods, such as Bayes Lasso Regression (BL), Bayes Ridge Regression (BRR), support vector machine (SVM).

5.1 Simulation study: phenotypes prediction

5.1.1 Simulations for the regression problems

To evaluate the performance of random Fourier features and random wavelet features, we refer to the simulation designs for predicting phenotypes from genotypes [29]. We assume the total proportion of variance in phenotype explained by genetic effects is 0.6, i.e. PVE = 0.6. We divide the genetic effects into two groups: (1) additive effects; (2) interaction effects. The additive effects make up $\rho\%$ of the genetic effects and the interaction effects make up the remaining $1 - \rho\%$.

Next we generate a matrix X with $n = 500$ observations and $p = 5000$ single-nucleotide polymorphisms (SNPs) and a vector y which represents the corresponding continuous phenotypes. We sample 30 causal SNPs and separate them into 15 additive SNPs and 15 interaction SNPs. We formulate the following simulation model

$$y = X\beta + Z\gamma + \epsilon,$$

where $\beta \sim \mathcal{N}(0, 3I)$, $\gamma \sim \mathcal{N}(0, 3I)$, $\epsilon \sim \mathcal{N}(0, I)$. Z is the genotype matrix for all pairs of interaction effects. ϵ is the random error.

Consider two scenarios depending on the parameter ρ since ρ indicates the proportion of different groups of the genetic effects. We choose ρ from the set $\{0.25, 0.75\}$. We set up the parameters in (4.4) and iterate the Gibbs sampler 2000 times. We compare our methods with two other standard Bayesian methods for regression: (1) Bayesian Ridge Regression [30] (2) Bayesian Lasso Regression [31]. The results are shown in Table 5.1, Figures 5.1 and 5.2.

Table 5.1: Comparison of the mean square error (MSE) for Bayes Ridge (BRR), Bayes Lasso (BL), Bayesian approximate kernel methods with Fourier and wavelet transform approximating Gaussian kernels with the parameter $\sigma = 1$. Values in bold represent the method with the lowest average MSE. Standard deviation (SD) for the replicates of each model is given in the parentheses.

	Methods	**MSE (SD)**
$\rho = 0.25$	Fourier transform	1.000 (0.190)
	Wavelet transform	**0.953** (0.086)
	Bayesian Ridge	1.118 (0.194)
	Bayesian Lasso	1.145 (0.157)
$\rho = 0.75$	Fourier transform	0.982 (0.130)
	Wavelet transform	**0.945** (0.158)
	Bayesian Ridge	1.085 (0.162)
	Bayesian Lasso	1.079 (0.232)

From Table 5.1, we conclude that the Bayesian approximate kernel method approximated by wavelet transform performs best among these four methods. The average mean square error is 0.953 for $\rho = 0.25$ with the smallest standard deviation of 0.086. For $\rho = 0.25$, wavelet transform also performs best with the MSE 0.94 and the standard deviations of our Bayesian approximate kernel methods are lower than those of

Bayesian Ridge and Bayesian Lasso methods. The Boxplots in Figures 5.1 and 5.2 agree with our conclusion that wavelet transform has the smallest error and standard deviation the Bayesian approximate kernel methods are more stable than the other Bayesian regression methods we use.

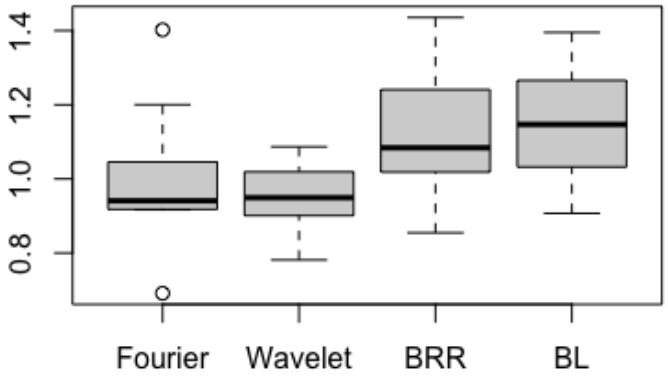

Figure 5.1: Boxplots for different methods when $\rho = 0.25$.

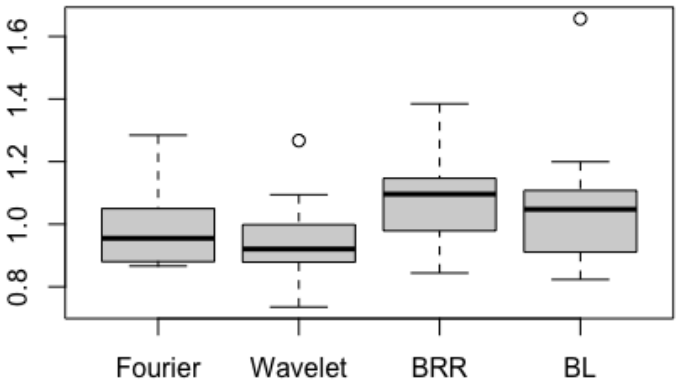

Figure 5.2: Boxplots for different methods when $\rho = 0.75$.

5.1.2 Simulations for the classification problems

We generate a matrix X with $n = 500$ observations and $p = 4000$ SNPs and a vector y which represents the corresponding class of phenotypes. Here $y_i = \{0, 1\}$ representing two categories of phenotypes. We conduct our experiment under the scenario $\rho = 0.75$. The other settings of the classification simulation are similar to those of the regression problems. We compare our Bayesian approximate kernel methods with the Support Vector Machine (SVM) [32] method in this section. We choose the Gaussian kernel as the kernel function in the SVM algorithms. We use accuracy to evaluate the models of classification. The accuracy is defined as

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}}$$

Table 5.2: Comparison of the accuracy and standard deviation for the Bayesian approximate kernel methods with Fourier and wavelet transform and Support Vector Machine (SVM) methods. The value in bold represents the method with the highest accuracy.

Methods	**Accuracy**	**Standard Deviation**
Fourier transform	0.583	0.049
Wavelet transform	**0.602**	0.022
SVM	0.471	0.035

From Table 5.2, we conclude that the Bayesian approximate kernel method approximated by wavelet transform has the highest accuracy 0.602 and the lowest standard deviation 0.022 among the three methods. Figure 5.3 shows that the results of the wavelet transform are more concentrated than those of the Fourier transform.

In a word, the Bayesian approximate kernel method approximated by wavelet transform has a good performance in both regression and classification simulations.

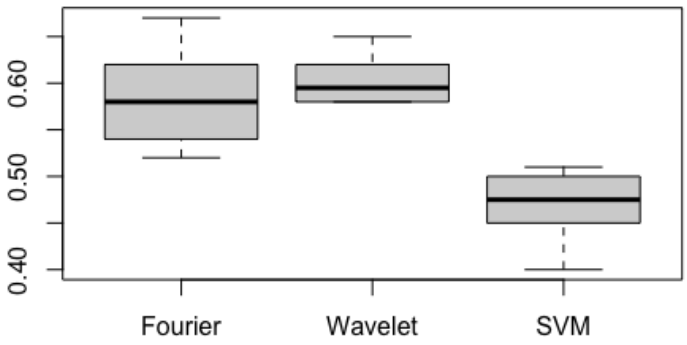

Figure 5.3: Boxplots for different classification methods.

5.2 Real data study

5.2.1 Real data for the regression problems

We further evaluate and compare the Bayesian approximate kernel method with Fourier and wavelet transform by analyzing the biscuit dough piece dataset from the R package functional data sets (fds) [33]. This example uses the near-infrared reflectance (NIR) spectra to measure the composition of biscuit dough pieces. The NIR spectrum of the observations is continuous curves, as shown in Figure 5.4. The information from these curves can be used to predict the composition of the biscuit. The compositions of the biscuit we estimated include fat, sucrose, flour and water and they all record in percent. We treat them as response values. The dataset contains 32 observations with 700 features. The results are shown in Table 5.3, Figures 5.5 and 5.6.

We conclude that the kernel functions approximated by wavelet transform perform better than those approximated by Fourier transform in our real data study. To be specific, the Bayesian approximate kernel using wavelet transform has smaller average mean square errors for all four compositions of the biscuit, which are 0.401,

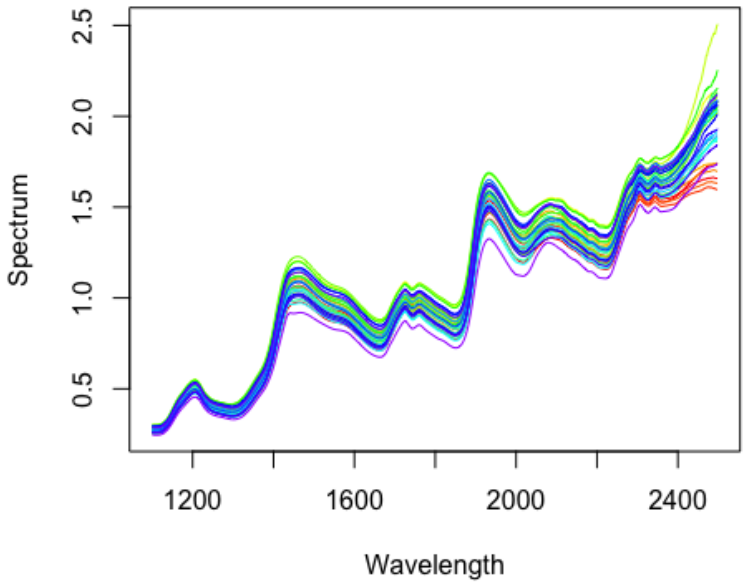

Figure 5.4: The NIR spectrum of the observations in the biscuit dough piece dataset.

Table 5.3: Comparison of the mean square error (MSE) for Bayesian approximate kernel methods with Fourier and wavelet transform approximating Gaussian kernels. Values in bold represent the method with the lower average MSE. Standard deviation (SD) for the replicates of each model is given in the parentheses.

Compositions	**Methods**	**MSE (SD)**
Fat	Fourier transform	0.459 (0.201)
	Wavelet transform	**0.401** (0.237)
Sucrose	Fourier transform	0.614 (0.271)
	Wavelet transform	**0.347** (0.144)
Flour	Fourier transform	0.526 (0.271)
	Wavelet transform	**0.348** (0.139)
Water	Fourier transform	0.387 (0.216)
	Wavelet transform	**0.378** (0.177)

0.347, 0.348 and 0.378. The standard deviations of Bayesian approximate kernel using wavelet transform are lower than those of the Bayesian approximate kernel

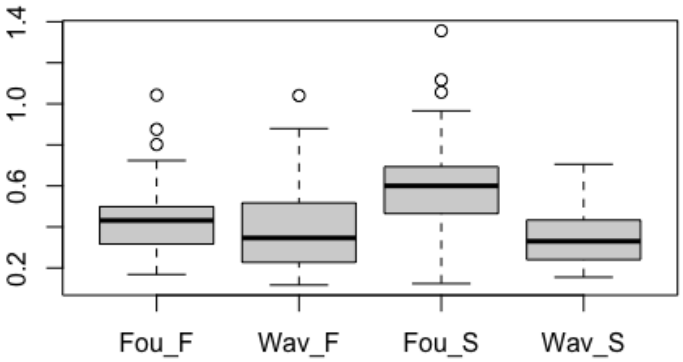

Figure 5.5: Boxplots of the Bayesian approximate kernel method with Fourier and wavelet transform results for the prediction of the fat and sucrose in the biscuit. **Fou_F** represents using Fourier transform for the prediction of the fat; **Wav_F** represents using wavelet transform for the prediction of the fat; **Fou_S** represents using Fourier transform for the prediction of sucrose; **Wav_S** represents using wavelet transform for the prediction of sucrose.

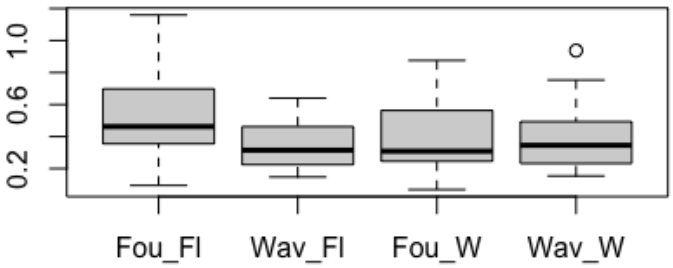

Figure 5.6: Boxplots of the Bayesian approximate kernel method with Fourier and wavelet transform results for the prediction of the flour and water in the biscuit. **Fou_Fl** represents using Fourier transform for the prediction of the flour; **Wav_Fl** represents using wavelet transform for the prediction of the flour; **Fou_W** represents using Fourier transform for the prediction of water; **Wav_W** represents using wavelet transform for the prediction of water.

using Fourier transform. These results are consistent with our simulation studies.

5.2.2 Real data for the classification problems

In this real data study for the classification problems, we use the Duke Breast Cancer database that consists of 86 tumour samples and 7129 genes. The data is numerical and has no missing values. The aim is to classify these tumour samples into estrogen receptor-positive (ER+) and estrogen receptor-negative (ER-) [34]. We can access the dataset from the following website: https://www.kaggle.com/andreicosma/duke-breast-cancer-dataset.

We compare the results of three methods applied to the Duke Breast Cancer dataset: (1) Bayesian approximate kernel approximated by Fourier transform; (2) Bayesian approximate kernel approximated by wavelet transform; (3) Support Vector Machine (SVM) with Gaussian kernel. The results are shown in Table 5.4 and Figure 5.7. We conclude that the Bayesian approximate kernel method approximated by wavelet transform performs the best among the three methods with an accuracy of 0.957. It is stable and accurate to use random wavelet features to approximate kernel function. SVM, which is the traditional method for nonlinear classification has the lowest accuracy of 0.55 processing the large p small n dataset.

Table 5.4: Comparison of the accuracy and standard deviation of three methods for the Duke breast cancer dataset. The value in bold represents the method with the highest accuracy.

Methods	**Accuracy**	**Standard Deviation**
Fourier transform	0.934	0.072
Wavelet transform	**0.957**	0.063
SVM	0.550	0.137

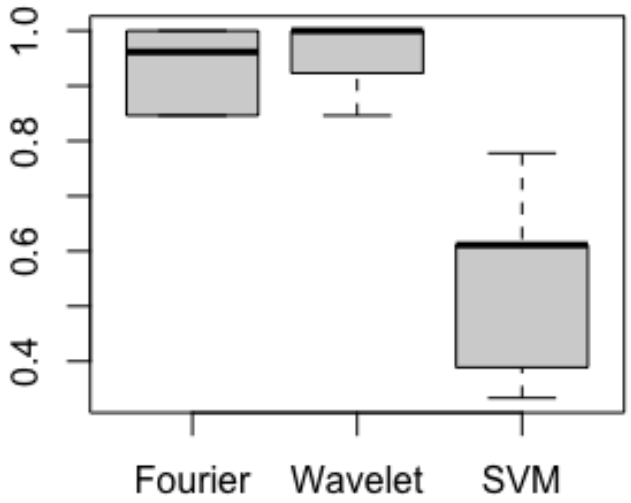

Figure 5.7: Boxplots of three classification methods for the Duke breast cancer dataset.

Chapter 6

Conclusions

This article proposes the Bayesian approximate kernel approximated by wavelet transform based on the framework of Bayesian approximate kernel regression [1]. We combine wavelet analysis with random features and use random wavelet features to approximate the kernel functions. The randomized feature map can lower the dimension. It is an efficient and computationally fast approach to deal with the large p small n problem. The method proposed in [1] uses random Fourier features. We use random wavelet features to improve the method since the wavelet transform is well localized in the both time and frequency domain. The performance of the kernel approximated by wavelet transform is better than that of the kernel approximated by Fourier transform when the data has big fluctuations in small intervals. We apply our method to both regression and classification problems and compare the performance with that of other classical methods.

Numerical studies prove that the Bayesian approximate kernel approximated by wavelet transform outperforms the Bayesian approximate kernel approximated by Fourier transform. We have smaller mean square errors solving regression problems and higher accuracy solving classification problems when using random wavelet features to approximate the kernel function. It shows that the random wavelet features method is stabler since its standard deviation of duplicates is smaller. We also conclude that the Bayesian approximate kernel methods perform better than other

Bayesian regression methods for the large p small n problems.

In conclusion, the Bayesian approximate kernel approximated by wavelet transform has a good performance in both regression and classification problems.

Bibliography

[1] L. Crawford, K. C. Wood, X. Zhou, and S. Mukherjee, "Bayesian approximate kernel regression with variable selection," *Journal of the American Statistical Association*, vol. 113, no. 524, pp. 1710–1721, 2018.

[2] F.-K. Wang and T. Du, "Using principal component analysis in process performance for multivariate data," *Omega*, vol. 28, no. 2, pp. 185–194, 2000.

[3] S. Chakraborty, M. Ghosh, and B. K. Mallick, "Bayesian nonlinear regression for large p small n problems," *Journal of Multivariate Analysis*, vol. 108, pp. 28–40, 2012.

[4] N. Aronszajn, "Theory of reproducing kernels," *Transactions of the American mathematical society*, vol. 68, no. 3, pp. 337–404, 1950.

[5] V. N. Vapnik, "An overview of statistical learning theory," *IEEE transactions on neural networks*, vol. 10, no. 5, pp. 988–999, 1999.

[6] Z. Zhang, G. Dai, and M. I. Jordan, "Bayesian generalized kernel mixed models," *The Journal of Machine Learning Research*, vol. 12, pp. 111–139, 2011.

[7] D. DeCoste and D. Mazzoni, "Fast query-optimized kernel machine classification via incremental approximate nearest support vectors," in *ICML*, 2003, pp. 115–122.

[8] A. Rahimi, B. Recht, *et al.*, "Random features for large-scale kernel machines.," in *NIPS*, Citeseer, vol. 3, 2007, p. 5.

[9] M Sifuzzaman, M. R. Islam, and M. Ali, "Application of wavelet transform and its advantages compared to fourier transform," 2009.

[10] B. K. Mallick, D. Ghosh, and M. Ghosh, "Bayesian classification of tumours by using gene expression data," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 67, no. 2, pp. 219–234, 2005.

[11] G. Wahba, *Spline models for observational data*. SIAM, 1990.

[12] J. Friedman, T. Hastie, R. Tibshirani, *et al.*, *The elements of statistical learning*, 10. Springer series in statistics New York, 2001, vol. 1.

[13] B. Schölkopf, R. Herbrich, and A. J. Smola, "A generalized representer theorem," in *International conference on computational learning theory*, Springer, 2001, pp. 416–426.

[14] E. G. Băzăvan, F. Li, and C. Sminchisescu, "Fourier kernel learning," in *European Conference on Computer Vision*, Springer, 2012, pp. 459–473.

[15] Z. Zhang, G. Wu, and E. Y. Chang, "Semiparametric regression using student t processes," *IEEE Transactions on Neural Networks*, vol. 18, no. 6, pp. 1572–1588, 2007.

[16] W. Rudin, *Fourier analysis on groups*. Courier Dover Publications, 2017.

[17] M. S. Bingham and K. R. Parthasarathy, "A probabilistic proof of bochner's theorem on positive definite functions," *Mathematics and computer science publications of Rennes*, pp. 1–13, 1967.

[18] N. Zhang and S. Ding, "Unsupervised and semi-supervised extreme learning machine with wavelet kernel for high dimensional data," *Memetic Computing*, vol. 9, no. 2, pp. 129–139, 2017.

[19] V. Vapnik, *The nature of statistical learning theory*. Springer science & business media, 2013.

[20] L. Zhang, W. Zhou, and L. Jiao, "Wavelet support vector machine," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 34, no. 1, pp. 34–39, 2004.

[21] H.-C. Shyu and Y.-S. Sun, "Construction of a morlet wavelet power spectrum," *Multidimensional systems and signal processing*, vol. 13, no. 1, pp. 101–111, 2002.

[22] J. M. Bernardo and A. F. Smith, *Bayesian theory*. John Wiley & Sons, 2009, vol. 405.

[23] J. A. Nelder and R. W. Wedderburn, "Generalized linear models," *Journal of the Royal Statistical Society: Series A (General)*, vol. 135, no. 3, pp. 370–384, 1972.

[24] S. Chakraborty, "Bayesian binary kernel probit model for microarray based cancer classification and gene selection," *Computational Statistics & Data Analysis*, vol. 53, no. 12, pp. 4198–4209, 2009.

[25] S. M. Lynch, *Introduction to applied Bayesian statistics and estimation for social scientists*. Springer Science & Business Media, 2007.

[26] J. Bernardo, P. Bernardo, E. Valencia International Meeting on Bayesian Statistics (2e : 1983 : Alcoceber, M. De Goot, D. Lindley, and A. Smith, *Bayesian Statistics 2: Proceedings of the Second Valencia International Meeting, September 6/10, 1983*, v. 2. North-Holland, 1985, pp. 371–372.

[27] S. Banerjee, B. P. Carlin, and A. E. Gelfand, *Hierarchical modeling and analysis for spatial data*. Chapman and Hall/CRC, 2003.

[28] J. H. Albert and S. Chib, "Bayesian analysis of binary and polychotomous response data," *Journal of the American statistical Association*, vol. 88, no. 422, pp. 669–679, 1993.

[29] X. Zhou, P. Carbonetto, and M. Stephens, "Polygenic modeling with bayesian sparse linear mixed models," *PLoS genetics*, vol. 9, no. 2, e1003264, 2013.

[30] A. E. Hoerl and R. W. Kennard, "Ridge regression: Biased estimation for nonorthogonal problems," *Technometrics*, vol. 12, no. 1, pp. 55–67, 1970.

[31] T. Park and G. Casella, "The bayesian lasso," *Journal of the American Statistical Association*, vol. 103, no. 482, pp. 681–686, 2008.

[32] W. S. Noble, "What is a support vector machine?" *Nature biotechnology*, vol. 24, no. 12, pp. 1565–1567, 2006.

[33] P. J. Brown, T. Fearn, and M. Vannucci, "Bayesian wavelet regression on curves with application to a spectroscopic calibration problem," *Journal of the American Statistical Association*, vol. 96, no. 454, pp. 398–408, 2001.

[34] M. West, C. Blanchette, H. Dressman, E. Huang, S. Ishida, R. Spang, H. Zuzan, J. A. Olson, J. R. Marks, and J. R. Nevins, "Predicting the clinical status of human breast cancer by using gene expression profiles," *Proceedings of the National Academy of Sciences*, vol. 98, no. 20, pp. 11 462–11 467, 2001.

Appendix A: Code

A.1 Section 1

Listing A.1: This is the code for random wavelet features

```
// [[Rcpp::export]]
arma::mat ApproxWaveletKernel3(arma::mat X, double iter) {
  double samp_size = X.n_cols;
  mat zeta(iter,samp_size);
  for(int i = 0; i<iter; i++){
    for (int j = 0; j < samp_size; j++) {
      vec n = randn(1); // b/a
      vec m = randn(1); // 1/a
      zeta(i, j) = 1;
      for (int k = 0; k < samp_size; k++) {
        zeta(i, j) *= cos(1.75*(as_scalar(m)*X(i,k)-as_scalar(n)))*
          exp(-pow((as_scalar(m)*X(i,k)-as_scalar(n)),2)/2);
      }
    }
  }
  mat K_hat = zeta.t()*zeta;
  return K_hat;
}

int rangeRand(int min, int max) {
  int range = max - min;
  return as_scalar(rand() % range + min);
}
}
```